Strategies of Traditional Chinese Character Recognition in Streetscape Based on Deep Learning Networks

Sin-Wun Syu and Po-Chyi Su*
*Dept. of Computer Science and Information Engineering, National Central University, Taiwan
E-mail: pochyisu@csie.ncu.edu.tw Tel/Fax: +886-3-4227151x35314

Abstract—Text recognition is an important task for extracting information from imagery data. Scene text recognition is one of its challenging scenarios since characters may have diversified fonts or sizes, be occluded by other objects and be captured from varying angles or under different light conditions. In contrast to alpha-numerical characters, Traditional Chinese Characters (TCC) receive less attention and the large number of TCC makes it difficult to collect and label enough scene-text images. This research aims at developing a set of strategies for TCC recognition. A synthetic dataset using a variety of data augmentation methods is constructed to simulate what may happen in real scenes, including text deformations, noise adding and background changes. A segmentation-based text spotting scheme is employed to locate the areas of text-lines and single characters. The characters can be recognized by the trained model and then linked into meaningful text-lines. The experimental results show that the proposed strategies work better in recognizing TCC in streetscape, when compared with existing publicly available tools.

I. INTRODUCTION

Text recognition is helpful to extract information from images or videos. Existing work can be divided into two categories based on the appearance of texts, i.e., document text recognition and scene text recognition (STR). STR is considered a relatively difficult task since the text in natural scenes may have diverse fonts or sizes, be occluded by other objects and be captured from varying angles or under different light conditions. Thanks to the availability of fast computing facilities, including Graphics Processing Units (GPU), and a massive amount of imagery data, deep learning is considered the mainstream solution nowadays as the deep neural networks can be trained to deal with varying cases. For text recognition, a large portion of current research focuses on alpha-numerical characters. On the other hand, Traditional Chinese Characters (TCC) receive less attention and labelled TCC datasets are also lacking. In fact, the large number of TCC further makes it difficult to collect and label images for training. The data imbalance issue becomes serious as some characters may have more real images than others. These challenges motivate us to investigate effective strategies for recognizing TCC.

It is worth noting that the methodologies of scene text recognition can be classified into two categories; one is segmenting single characters and then recognizing them one by one, usually via convolutional neural networks (CNN), and the other one is employing recurrent neural networks (RNN) to convert horizontal text-lines into sequences with multiple characters as the output. RNN-based methods seem more popular these days, especially when applied to the recognition of English letters. In this research, nevertheless, the CNN-based methodology is considered for the following reasons. First, as we know, each Chinese character has an independent meaning and the relevance of characters in a text-line is lower than that of letters in English sequences. The advantages provided by RNN-based methods may not be that obvious. Second, the text-line in Chinese may appear horizontally, vertically or even curved in streetscape. The text-line recognition using RNN-based methods usually requires additional pre-processing to modify the shapes of text-lines. Third, recognizing single characters can simplify this text recognition task into a classification problem using CNN, and the training speed is faster than RNN-based methods. Fourth, it is easier to see which character could be wrongly recognized and no need to deal with duplicated characters that may happen in RNN-based methods.

To effectively recognize TCC, a set of strategies for recognizing TCC in streetscape are proposed in this research. We developed a synthetic dataset using several data augmentation approaches, including text deformations, noise adding and background changes, to simulate the cases in real scenes. A segmentation-based text spotting scheme is used to locate the areas of text-lines and characters so that characters can be recognized by the trained model and then linked into meaning text-lines. The STR network is designed to enhance the recognition performance of the model by appropriate data pre-processing and model training. The remaining of the paper is organized as follows. Sec. II reviews the related work of detection and recognition of scene texts. Sec. III details the proposed scheme. The experimental results shown in Sec. IV demonstrate the feasibility of the proposed scheme. Finally, the conclusion and future work are discussed in Sec. V.

II. RELATED WORK

Many existing methods based on deep learning have been proposed to detect and recognize scene texts. For example, CRNN[1] is often used to output sequences with varying lengths of characters using Connectionist Temporal Classification (CTC) loss function[2]. Although CTC allows to predict a character at each column and modify the full character sequence into a flexible stream of characters by deleting repeated characters and blanks, it has poor results

on skewed texts. RARE[3] proposed a transformation module that can rectify the input into a straight text image. SAR[4] has had great success on skewed text appearance via 2-D attention mechanism without losing spatial information. Mask TextSpotter[5] treated the recognition task as a semantic segmentation problem for spotting text with arbitrary shapes. CRAFT[6] produced the region score and affinity score to localize individual characters and group each character into a single instance. PP-OCR[7] employed DB[8] as the text detector, using CRNN as the text recognizer, and proposed a light-weight text recognition system by adjusting the backbone and training strategies. NRTR[9] was based on Transformer[10]. The model has the same characteristics as RNN, but it can be operated in parallel, which greatly reduces the training time, and improve the accuracy of the model through stacked self-attention.

In CNN-based character recognition, [11] proposed a video OCR scheme, including text detection, character segmentation, CNN-based character recognition and correct text using NLP model. For the related research of Simplified Chinese, CTW[12] contained massive character-level annotations for training CNN networks to achieve effective recognition.

III. PROPOSED METHOD

The proposed scheme is divided into two main parts: scene text detector and rectified character recognizer. The scene text detection network helps to extract the locations of text-lines and individual characters. The scene text recognition network is then employed to identify each character. After re-arranging the characters according to their coordinates in the image, meaningful text-lines can be formed. For the design of STR network, we further subdivide it into three steps: (1) the synthetic training dataset, (2) the data augmentation methods and (3) the design of network recognition.

A. Scene text detector

Scene text detection methods based on deep learning can be roughly classified into two types: pixel-based segmentation and region-based object detection. Each methodology has its advantages and disadvantages. This research employs [13] as our scene text detector, which combines the characteristics of these two types of networks, i.e., a main network architecture based on semantic segmentation supplemented by a refining network based on region-based object detection. Non-maximum suppression (NMS) is then applied to filter some errors to achieve a better outcome.

In this research, we trained two networks to extract text-lines and individual characters. The text-line detection network uses ICDAR 2017 as the training data, which provides not only text-line labels, but also a small number of images containing Chinese characters. The character detection network uses CTW dataset[12] containing a large number of Simplified Chinese character-level annotations. Figure 1 shows our detection results, from which visible characters or text-lines can be extracted successfully. The region identified by the character detection network can be reorganized according to the text-line regions with the writing rule of ”from left to right, from top to bottom”. We can thus combine the recognized characters into meaningful text-lines following this arrangement.

Fig. 1. Scene text detection results: (a) Original image; (b) Text-line detection results; (c) Character detection results

B. Rectified character recognizer: synthetic dataset

Individual Chinese characters are the target of recognition in the proposed scheme. Classification candidates, i.e. Chinese characters, have to be defined so that the dataset can be constructed accordingly. The number of Chinese characters is as high as tens of thousands so selecting a suitable subset containing common characters should be preferred. We chose 4808 Chinese characters by referring [14] and added some simplified Chinese characters that can sometimes be seen in streetscape. Totally 5028 characters are the candidates. Excluding rare characters helps the model to perform better in average.

In view of the fact that it is not easy to find suitable TCC scene-text datasets and that collecting and labeling a massive amount of imagery data by ourselves is a tiresome work, we developed a synthetic dataset with automatic labeling, which avoids the problems of incorrect labeling and imbalanced categories. Making a synthetic dataset starts with constructing varying images for each character using different font files. We collected 152 different Chinese font files. The grayscale characters are formed by processing the font files through Python image library (PIL) and Unicode. The image of each character has the size 64×64 with black strokes and white background. Some examples are shown in Figure 2. The generated images are examined to remove empty ones, which means that some font files may not contain certain characters. $764,256$ images are generated to form the synthetic dataset, in which $611,405$ images are used as the training set and the remaining $152,851$ images are used as the validation set.

Fig. 2. Examples of printed characters in the synthetic dataset

C. Rectified character recognizer: data augmentation

The images in the synthetic dataset are printed characters with different fonts. Data augmentation is required to twist each character image to generate varying forms. MjSynth[15] is referred to produce more synthetic data for training. Given one image showing a single character, one or multiple modifications selected from 15 data augmentation methods will be applied. The data augmentation methods include text deformation and background changes. Some examples of text deformation and background changing are shown in Fig. 3 and Fig. 4 respectively.

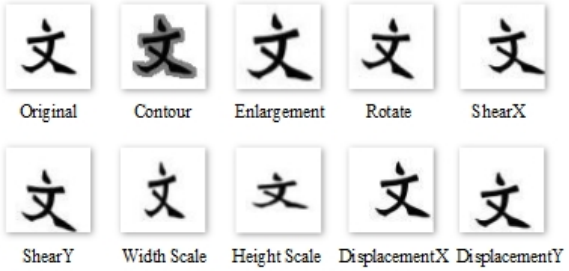

Fig. 3. Data augmentation for text deformation

Fig. 4. Data augmentation for background changes

Although some synthetic images after data augmentation may not conform to what texts appear in real scenes, we believe that the variety of synthetic data is beneficial to the model training, which can make the model more adaptive to the ever-changing environment. During the training, we refer to the method of RandAugment[16], and set N as the number of data augmentation types for each invoke. $N = 1$ represents a random data augmentation for an image, $N = 2$ stands for two kinds of data augmentation applied on the image. The larger N is, the more complex an image becomes. Figure 5 shows the effects of using different N. Different from RandAugment, we do not use a fixed N, but gradually increase N from 1 to 7 in each epoch. We found that this strategy can make the model learn from simple images to difficult ones.

Fig. 5. The effects of N kinds of data augmentation

D. Rectified character recognizer: STR network design

The feature normalization is used first to scale the feature data to a certain range as it helps to make the data have a better structure, resulting in faster convergence in the gradient descent calculation and more accurate recognition as well. To be more specific, the mean and standard deviation of all data are calculated first and then each input data are normalized to generate more suitable output. The semantic and spatial information of the extracted feature map will affect the classification performance of the deep network. Properly deepening and widening the feature network are thought to be helpful to the model. As we treat the recognition of TCC as a multi-classification problem, after thorough evaluation, we chose ResNeSt[17] as our network architecture as it presents a split-attention block structure, which performs better among all existing ResNet variants and has similar computational efficiency. In our implementation, the input channel is modified to 1 and the output of the last fully connected layer is modified to 5028 to adapt to our grayscale training images and text classification categories. In addition, we evaluated ResNeSt-50 and ResNeSt-101, and we found that the accuracy of the latter (deep) model is actually lower than the former one. This observation indicates that deepening the depth of the network may increase the training difficulty but is not beneficial to the TCC recognition. Therefore, ResNeSt-50 is chosen as the backbone network.

IV. EXPERIMENTAL RESULTS

As there is no publicly available Traditional Chinese dataset that can be used as a test set, we took pictures by ourself and cropped the areas containing TCC from the pictures for testing. Some examples are shown in Figure 6. $29,284$ street-view images containing single characters were collected and labeled as the test-set to evaluate the accuracy. The experimental results show that the accuracy is as high as 91.00%.

Fig. 6. Example of "character" test-set

Since our training data are all synthetic data, there still exists room for improvement. We employed the Google API service to collect a large number of street-view images. First, a random initial coordinate is given, and the coordinates of these landmarks are obtained using Google Place API to search the nearby landmarks of the initial coordinate. Then the Street View Static API is used to take photos of these locations, and we continue to search the nearby landmarks with the Google Place API. Following these steps, an automatic process is designed to collect a large number of street-view images by looping these steps. After the process of text detection, recognition and correction, the proposed method extracts images of single characters from the street-view images, and labels them accordingly. Finally, $41,136$ street-view images containing single characters are collected. After collecting and labeling these data, we mixed it with the synthetic data to fine-tune the model weights, and used the previously mentioned street-view test-set containing single characters to conduct the experiments. After this fine-tuning processing, the accuracy can be raised to 93.31%.

We also used the street-view photos taken by ourselves to form the text-line test-set, in which related characters are linked into meaningful words. Some examples are shown in Figure 7. There are $4,023$ images, containing $8,189$ text-lines. We adopted the Normalized Edit Distance to evaluate the recognition results on text-lines as

$$accuracy = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{D\left(si, \hat{si}\right)}{max\left(si, \hat{si}\right)}, \tag{1}$$

where N represents the total number of text-lines, si is the identification result, $\hat{si}$ is the ground truth. $D(si, \hat{si})$ calculates the Edit Distance of the two strings, and $max(si, \hat{si})$ is the longer length of the two text-lines. Table I shows the results compared with the other two recognition methods. We can see that the proposed scheme outperforms CRNN and PP-OCR, with the accuracy reaches 0.87.

Method	Accuracy
CRNN[1]	0.76
PP-OCR[7]	0.82
Ours	0.87

TABLE I: The comparison of recognizing the text-line test-set

Fig. 7. Examples of "Text-line" test-set. The upper half shows the original images, and the lower half shows the labeled text-area, where the green box represents characters, and the red box represents text-lines.

Finally, we compared the proposed scheme with two publicly available tools, Google Vision and Line OCR. Figure 8 shows one typical case. Google vision predicted「鹽疆錄」, in which none of the words are correct. Line OCR predicted「鉅大摧球」, in which the third charcter is wrong. In our case, as we added contours around the texts and applied blurring on the background in the process of data augmentation, the four characters 「鉅大撞球」can be recognized correctly, even though the text style is quite special.

Fig. 8. The comparison of the proposed scheme and two publicly available tools: (a) Original image, (b) our result, (c) Google Vision, and (d) Line OCR.

V. CONCLUSION

In this research, the traditional Chinese scene text recognition strategies based on deep learning is proposed, which can effectively recognize TCC in street-view images and improve the overall recognition accuracy of text-lines linked by char-

acters through a text-line correction mechanism. Compared with other existing publicly available tools, the proposed scheme has better performance. As this research only focuses on the TCC recognition, some of the text-lines in scenes mixed with alphanumeric characters and Chinese may not be recognized completely. If a large number of training materials of alphanumeric characters (in other languages) are added in the future, the scope of applications can become broader. In addition, more street-view imagery data is definitely helpful to keep us fine-tuning the model and improve the recognition accuracy.

ACKNOWLEDGMENT

This research is supported by the Ministry of Science and Technology in Taiwan, under Grants MOST 109-2221-E-008-076 and 110-2221-E-008-086.

References

[1] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 11, pp. 2298–2304, 2016.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in Proceedings of the 23rd international conference on Machine learning, pp. 369–376, 2006.

[3] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4168–4176, 2016.

[4] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8610–8617, 2019.

[5] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 67–83, 2018.

[6] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374, 2019.

[7] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang, et al., "Pp-ocr: A practical ultra lightweight ocr system," arXiv preprint arXiv:2009.09941, 2020.

[8] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474–11481, 2020.

[9] F. Sheng, Z. Chen, and B. Xu, "Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition," in 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 781–786, IEEE, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, pp. 5998–6008, 2017.

[11] K. Elagouni, C. Garcia, and P. Sébillot, "A comprehensive neural-based approach for text recognition in videos using natural language processing," in Proceedings of the 1st ACM International Conference on Multimedia Retrieval, pp. 1–8, 2011.

[12] T.-L. Yuan, Z. Zhu, K. Xu, C.-J. Li, and S.-M. Hu, "Chinese text in the wild," arXiv preprint arXiv:1803.00085, 2018.

[13] G.-X. Zeng, Y.-H. Hou, P.-C. Su, and L.-W. Kang, "Scene text-line extraction with fully convolutional network and refined proposals," in 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1247–1251, IEEE, 2020.

[14] "教育部成果網-常用字下載." https://language.moe.gov.tw/result.aspx?classify_sn=23&subclassify_sn=437&content_sn=46. Accessed: 2021-06-22.

[15] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," arXiv preprint arXiv:1406.2227, 2014.

[16] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical data augmentation with no separate search," arXiv preprint arXiv:1909.13719, vol. 2, no. 4, p. 7, 2019.

[17] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, et al., "Resnest: Split-attention networks," arXiv preprint arXiv:2004.08955, 2020.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, Ieee, 2009.

[19] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?," arXiv preprint arXiv:1906.02629, 2019.